

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian Sebelumnya memberikan kesimpulan bahwa metode pengembangan sistem melalui pendekatan *Rapid Application Development* atau RAD dapat memberikan pengaruh pada *development speed* dengan hasil aplikasi berkualitas unggul dan biaya yang murah. Integrasi dengan aplikasi eksternal juga memberikan kemudahan mengolah data, sinkronisasi data, dan menyajikan informasi yang membuat proses bisnis menjadi lebih mudah dan efisien. Pemilihan *framework* Next.js dipilih karena terbukti dengan menghasilkan aplikasi yang berkualitas dalam kategori kinerja, kecepatan, aksesibilitas, dan juga SEO. Metode pengujian *Black-box Testing* digunakan untuk menguji fungsionalitas aplikasi tanpa memperhatikan kualitas dari code yang disusun karena fokus dari aplikasinya adalah penggunaan aplikasi tanpa ada kendala yang ditemukan dari sisi *user*. *System Usability Scale* (SUS) juga digunakan untuk mengevaluasi sejauh mana suatu sistem atau aplikasi dapat digunakan dengan efektif dan memuaskan oleh pengguna nantinya.

Penelitian ini ditunjang dengan mempelajari beberapa jurnal terdahulu dengan memakai teknik 3C2S. 3C2S ini adalah singkatan dari *Comparing, Contrasting, Criticize, Synthesize, Summarize*. Berikut adalah daftar penelitian sebelumnya yang disajikan pada tabel 2. 1.

Tabel 2. 1 Penelitian Sebelumnya

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
1	Empowering Efficiency: A Web-Based Inventory and Sales Information System for Drinking Water Distributors through Rapid Application Development[13]	2022	Penelitian ini menggunakan pendekatan metode RAD sebagai model perancangan sistem dan metode <i>Black Box</i> sebagai metode pengujiannya. Sistem yang dibuat pada penelitian ini adalah POS (<i>Point of Sale</i>).	Metode RAD pada penelitian ini terbukti cocok untuk mengembangkan sistem informasi yang mengutamakan efisiensi.	Penelitian ini hanya menggunakan dua model UML yaitu <i>use case diagram</i> dan <i>activity diagram</i> . Penelitian ini tidak menyebutkan dengan alat apa sistem akan dibuat.	Penelitian ini menjelaskan dengan lengkap setiap tahapan dari model pengembangan RAD.	Penelitian ini berfokus pada UMKM distribusi air minum. Model pengembangan sistem yang dipilih menggunakan metode RAD dan pengujiannya menggunakan metode <i>Black Box Testing</i> . Penerapan metode RAD menunjukkan dapat meningkatkan efisiensi dalam siklus waktu yang terbatas.
2	Perancangan Sistem Informasi Fasilitas Fitness Gym Berbasis Website Menggunakan Codeigniter[14]	2023	Penelitian ini merancang sistem informasi termasuk keanggotaan menggunakan <i>framework</i>	Penelitian ini berfokus pada pengembangan dari sistem lama menuju sistem baru yang terkomputerisasi.	Penelitian ini tidak mencantumkan pembuatan diagram UML dalam perancangan sistemnya.	Penelitian ini menjelaskan bahwa sistem baru ini meningkatkan efisiensi dari	Penelitian ini tidak secara jelas menyebutkan pendekatan metode pengembangan sistem yang

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
			Codeigniter.			pada menggunakan sistem manual sebelumnya dalam pendataan anggota.	digunakan. <i>Framework</i> yang digunakan untuk merancang sistem menggunakan Codeigniter dan untuk perancangan sistemnya menggunakan metode waterfall.
3	Penerapan Metode <i>Rapid Application Development</i> dalam Perancangan Sistem Informasi Pendataan[15]	2020	Model Rapid Application Development adalah pendekatan yang diterapkan sebagai metode perancangan pada jurnal ini. Bahasa pemrograman yang digunakan untuk perancangan sistem yaitu PHP murni.	Metode RAD yang digunakan mengikuti pada versi <i>requirements planning, design workshop, dan implementation.</i>	Penelitian ini tidak menggunakan metode pengujian sistem sehingga tidak diketahui bagaimana performa dari aplikasinya.	Penelitian ini menunjukkan bahwa penggunaan sistem informasi pencatatan anggota dapat memudahkan penyajian dan pengelolaan data untuk kepentingan organisasi.	Penelitian ini menggunakan pendekatan RAD untuk metode perancangan sistemnya. Aplikasi pada penelitian ini dirancang menggunakan PHP murni sebagai bahasa pemrogramannya. Hasil pada penelitian tersebut menunjukkan bahwa sistem

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
							informasi pendataan ini memudahkan pengelolaan data anggota dan juga memudahkan penyajian datanya.
4	Perancangan Sistem Informasi Pengelolaan <i>Member Gym</i> Berbasis <i>Website</i> (Studi Kasus: Galby Gym Padalarang)[16]	2022	Model <i>prototype</i> digunakan pada penelitian ini sebagai metode perancangan sistemnya. Bahasa pemrograman PHP digunakan untuk merancang aplikasinya tanpa menggunakan <i>framework</i> .	Pengguna yang ada didalam aplikasi ini terdiri dari admin dan <i>owner</i> .	Penelitian ini tidak menggunakan metode pengujian sistem sehingga tidak diketahui bagaimana performa dari sistem yang telah dibuat.	Penelitian ini menunjukkan bahwa sistem informasi pengelolaan anggota dapat mempermudah dan mempercepat admin dalam melaksanakan tugasnya.	Penelitian ini menggunakan pendekatan <i>prototype</i> untuk perancangan sistemnya dan bahasa yang digunakan untuk merancang sistemnya adalah PHP murni. Akto yang terlibat didalam sistem informasi ini adalah admin dan <i>owner</i> . Hasil kesimpulan dari sistem informasi ini dapat mempermudah dan

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
							mempercepat admin dalam melaksanakan tugas pengelolaan data anggota.
5	Implementasi Website Full-Stack Menggunakan Teknologi Next.Js, React, Dan Sanity[17]	2023	Perancangan sistem pada penelitian ini menggunakan bahasa javascript dengan menggunakan library React Js dan framework Next Js.	Penelitian ini merancang sebuah sistem <i>fullstack</i> yaitu antara <i>backend</i> dan <i>frontend</i> tergabung dalam satu sistem utuh.	Penelitian ini tidak menyebutkan metode apa yang digunakan untuk perancangan sistemnya begitupun dengan pengujian yang digunakan tidak menyebutkan metode yang jelas.	Penggunaan teknologi Next Js menghasilkan skor yang tinggi berdasarkan alat pengukur performa web dari google yaitu <i>Lighthouse</i> .	Penelitian ini menunjukkan hasil sistem yang berkualitas tinggi berdasarkan skor dari <i>Lighthouse</i> dalam kategori kinerja, kecepatan, aksebilitas, dan juga SEO dengan menggunakan framework Next Js.
6	<i>Application of the Rapid Application Development Model to a Web-Based Library Information System</i> [18]	2023	Penelitian ini menggunakan pendekatan RAD sebagai metode pengembangan sistem dan bahasa pemrograman PHP murni digunakan sebagai alat	Metode RAD yang digunakan mengikuti pada versi <i>requirements planning, design workshop, dan implementation.</i>	Penelitian ini tidak menyebutkan pengaruh dari perancangan sistem yang menerapkan metode RAD.	Penelitian ini telah menerapkan <i>data modelling</i> yang lengkap dalam model UML.	Perancangan sistem pada penelitian ini menerapkan metode RAD dan Black Box Testing dipilih sebagai metode pengujian sistem pada penelitian tersebut.

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
			perancangan sistem informasi.				Hasil dari penelitian ini menyebutkan bahwa perancangan sistem informasi ini memudahkan dalam pencarian data dan membuat pelaporan perpustakaan lebih baik dan efisien.
7	<i>Implementation of Rapid Application Development Method in the Development of Geographic Information Systems of Industrial Centers</i> [19]	2020	Metode yang diterapkan pada penelitian ini menggunakan pendekatan RAD dan untuk merancang sistemnya menggunakan bahasa pemrograman PHP dengan <i>framework</i> Codeigniter	Metode RAD yang digunakan mengikuti pada versi <i>planning, analysis, design, dan implementation.</i>	Penelitian ini tidak menyebutkan metode untuk pengujian sistem. Tahap <i>design</i> pada penelitian ini tidak dicantumkan alur kerja sistemnya menggunakan UML.	Penelitian ini membuktikan bahwa metode RAD adalah metode yang cepat untuk menghasilkan sistem yang unggul.	Metode RAD digunakan pada perancangan sistem pada penelitian ini. Bahasa pemrograman untuk merancangnya menggunakan PHP dengan <i>framework</i> Codeigniter. Hasil dari penelitian ini membuktikan ketangguhan RAD dalam

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
							menghasilkan produk berkualitas unggul dengan jangka waktu lebih cepat. Sistem ini diselesaikan dalam waktu 84 hari atau 12 bulan.
8	<i>A comparative analysis of RAD and agile technique for management of computing graduation projects</i> [20]	2020	Penelitian ini membandingkan pro dan kontra dari penggunaan metode RAD dan Agile dalam proses tugas ahir mahasiswa.	Penelitian ini merangkum rincian perbandingan dari metodologi RAD dan Agile	Ukuran sampel kecil untuk studi kasus dan hanya dua penilaian yang dipilih untuk perbandingan.	Penelitian ini menemukan bahwa proyek yang dikembangkan dengan mengikuti metodologi RAD memberikan performa 13,33% lebih baik dalam menyediakan dokumentasi yang luas dan rinci dibandingkan	Penelitian ini Membuktikan bahwa untuk performa penyediaan dokumentasi RAD mendapatkan nilai lebih tinggi dari Agile. Metode RAD cocok untuk pengembangan project secara individu. Penggunaan RAD dalam proyek skala kecil mendapatkan manfaat dengan hasil yang lebih

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
						dengan mahasiswa yang mengikuti teknik Agile.	cepat didapatkan dan biaya lebih murah.
9	Perbandingan Metode <i>Waterfall</i> dan RAD (<i>Rapid Application Development</i>) pada Pengembangan Sistem Informasi[21]	2022	Penelitian ini memeberikan gambaran dari kelebihan dan kekurangan dari penggunaan metode RAD dan <i>Waterfall</i> .	Penelitian ini merangkum dari penelitian-penelitian terdahulu untuk diambil perbandingannya.	Penelitian ini tidak menggunakan alat uji untuk membuktikan kebenarannya.	Penggunaan metode RAD cocok digunakan dalam proyek skala besar karena siklus waktunya pendek dan menghemat biaya karena menggunakan konsep <i>iterative</i> .	Penelitian ini memeberikan gambaran dari kelebihan dan kekurangan dari penggunaan metode RAD dan <i>Waterfall</i> . Penggunaan RAD dalam pemilihan metode perancangan sistem jika terdapat perubahan bisa dengan mudah untuk dirubah dibandingkan dengan menggunakan <i>Waterfall</i> .
10	Rancang Bangun <i>Customer</i>	2023	Pendekatan yang digunakan pada	Penelitian ini menghasilkan	Penelitian ini tidak menyebutkan dengan	Sistem informasi yang	Hasil dari penelitian ini

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
	<i>Relationship Management</i> pada Factorylandstore[22]		penelitian ini yaitu menggunakan metode Waterall untuk perancangan sistemnya. Metode yang digunakan untuk pengujian sistem yang telah dibuat pada penelitian ini menggunakan metode <i>black box testing</i> .	sistem yang terintegrasi dengan aplikasi eksternal menggunakan API dari telegram.	bahasa pemrograman apa yang digunakan untuk mengintegrasikan layanan aplikasi kepada telegram.	dibuat berintegrasi dengan telegram sehingga menghasilkan informasi yang sesuai antara bot telegram maupun sistem CRM yang telah dirancang.	berupa sistem CRM yang terintegrasi dengan telegram melalui bot API. Integrasi dengan bot telegram ini mempermudah pelanggan dalam mendapatkan informasi terkait Factorylandstore.id.
11	Pengembangan Sistem Informasi Geografis Banjir di Kabupaten Klaten dengan Memanfaatkan Media <i>Website</i> dan WhatsApp[23]	2023	Metode yang digunakan pada penelitian ini adalah <i>Rapid Application Development</i> . <i>Framework</i> yang digunakan untuk perancangan sistem yaitu Laravel.	Penelitian ini menghasilkan aplikasi yang terintegrasi pada layanan eksternal yaitu WhatsApp.	Test case pada bagian pengujian <i>black-box testing</i> tidak dijelaskan semua bagian fungsionalitasnya.	Pengujian <i>black-box testing</i> diterapkan pada penelitian ini untuk menguji fungsionalitas sistem dan menggunakan SUS untuk menguji tingkat kepuasan pengguna.	Penelitian ini menghasilkan suatu aplikasi SIG atau sistem informasi geografis yang terintegrasi dengan whatsapp. Model pengembangan sistem menggunakan RAD serta pengujiannya menggunakan <i>black-box testing</i>

No.	Judul	Tahun	Comparing	Contrasting	Criticize	Synthesize	Summarize
							dan juga SUS yang menghasilkan nilai 70.02 dengan mendapatkan grade C kategori dapat diterima.
12	Rancang Bangun dan Implementasi Sistem Informasi Manajemen Penjualan Berbasis Website Menggunakan Kerangka Kerja CodeIgniter pada Kafe Dangau Kopi di Daerah Gunung Pati, Kota Semarang, Jawa Tengah[24]	2023	Metode <i>Rapid Application Development</i> diterapkan pada penelitian ini dengan <i>framework</i> CodeIgniter sebagai alat perancangannya.	Penelitian ini menghasilkan aplikasi POS yang memiliki <i>multilevel</i> user.	Penelitian ini tidak menjelaskan secara mendetail bagaimana alur pengerjaan sistemnya. Tahapan untuk user design juga tidak menggunakan diagram UML secara lengkap melainkan hanya menggunakan <i>use case diagram</i> .	Pengujian <i>balck-box testing</i> diterapkan pada penelitian ini untuk menguji fungsionalitas sistem dan menggunakan SUS untuk menguji tingkat kepuasan pengguna.	Hasil dari penelitian ini yaitu suatu aplikasi POS yang telah diuji dengan menggunakan blak-box testing dan juga SUS menghasilkan nilai 76 dan memperoleh grade C yang termasuk kategori <i>acceptable</i> .

2.2 Landasan Teori

2.2.1. Rancang Bangun

Rancang adalah proses dengan tujuan merancang *system* yang baru sehingga mampu mengatasi permasalahan pengguna, yang asalnya dari pemilihan opsi *system* yang paling optimal. Sementara itu, kata "bangun" merujuk pada aktivitas pembangunan yang mencakup penciptaan sistem baru atau perubahan serta perbaikan pada *system* terdahulu, baik secara menyeluruh maupun sebagian[25].

Rancang bangun adalah istilah yang sering digunakan dalam proses menyusun, merencanakan, serta merancang suatu objek atau sistem, mulai dari tahap awal pembuatan hingga tahap akhirnya. Hal ini meliputi serangkaian langkah yang mencakup penggambaran, perencanaan detil, serta pembuatan sketsa atau pengaturan dari berbagai elemen yang mungkin terpisah, sehingga dapat disatukan menjadi sebuah kesatuan yang lengkap, terpadu, dan berfungsi sesuai dengan tujuannya. Proses rancang bangun ini mempertimbangkan aspek-aspek teknis, estetika, fungsionalitas, dan keandalan untuk menciptakan produk atau sistem yang memenuhi kebutuhan serta memastikan kualitasnya secara menyeluruh[26].

Rancang bangun merujuk pada sebuah proses pengembangan sistem yang bertujuan untuk menciptakan sistem baru atau melakukan perubahan serta perbaikan pada sistem yang sudah ada, baik itu dilakukan secara menyeluruh maupun hanya pada bagian tertentu dari sistem tersebut[27].

2.2.2. Sistem Informasi

Sistem informasi adalah suatu kesatuan terdiri dari elemen-elemen atau komponen yang saling terhubung, berkolaborasi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi. Fokusnya adalah hubungan antar komponen dalam melakukan aktivitas seperti pengumpulan, pemrosesan, penyimpanan, dan distribusi informasi. Tujuannya adalah mendukung pengambilan keputusan, koordinasi, pengendalian, serta memberikan gambaran menyeluruh tentang kegiatan di organisasi. Sistem informasi juga berfungsi sebagai suatu langkah yang mengumpulkan, mengubah data menjadi informasi,

dan mendistribusikannya kepada pengguna. Secara keseluruhan, sistem informasi dapat dianggap sebagai rangkaian elemen yang bekerja bersama untuk mengelola informasi, dengan tujuan mendukung kegiatan operasional, manajerial, dan strategi suatu organisasi yang melibatkan pengolahan data terstruktur dengan hasil yang bermanfaat[26][27].

2.2.3. Keanggotaan

Keanggotaan/*Membership* merujuk pada tindakan memperoleh dukungan formal atau pengakuan sebagai individu yang tergabung dalam suatu kelompok. Saat seseorang diterima menjadi bagian dari suatu kelompok, hal ini menandakan pengakuan resmi terhadap posisinya dalam entitas tersebut. Keanggotaan dapat berkaitan dengan kelompok informal seperti unit keluarga, serta juga dapat terkait dengan entitas formal seperti organisasi agama, perkumpulan, persaudaraan, serikat mahasiswa, asosiasi perdagangan, atau kelompok tertentu. Ini mencerminkan hubungan resmi yang dibentuk antara individu dan entitas tersebut, dengan setiap kelompok memiliki aturan, norma, dan kewajiban yang biasanya diakui atau dijalankan oleh anggotanya[30].

2.2.4. Website

Web adalah sebagian dari *service* yang beroperasi pada infrastruktur teknologi internet. Halaman web menjadi dapat diakses melalui pemanfaatan teknologi *web server* sebagai penyedia utama halaman-halaman tersebut. Penggunaan bahasa baku seperti HTML menjadi fondasi yang memungkinkan tampilan dan struktur halaman web. Selain itu, pengiriman dokumen web terjadi melalui jalur komunikasi HTTP yang memungkinkan perpindahan data antara *server web* dan pengguna. Teknologi ini secara bersama-sama membentuk ekosistem yang memungkinkan akses dan interaksi dengan informasi melalui internet. Keberadaan teknologi-teknologi ini membuka jalan bagi pengembangan layanan web yang semakin kompleks dan responsif terhadap kebutuhan pengguna. Berdasarkan klasifikasinya terdapat dua jenis *website* yaitu *website* statis yang kontennya sulit diubah oleh pengguna dan memerlukan perubahan melalui kode

atau basis data, serta *website* dinamis yang dapat beradaptasi secara otomatis dengan perubahan konten tanpa mengubah struktur kode *website*[31][32].

Aplikasi berbasis web merujuk pada jenis aplikasi yang bisa diakses melalui jaringan internet atau intranet dengan menggunakan peramban atau *browser*. Aplikasi ini dirancang menggunakan berbagai bahasa pemrograman seperti HTML, JavaScript, CSS, PHP, dan lainnya. Kelebihan utamanya terletak pada aksesibilitas lintas *platform*, memungkinkan informasi dari aplikasi web diakses melalui bermacam-macam sistem operasi melalui web *browser*. Aplikasi ini memiliki fleksibilitas penggunaan untuk berbagai tujuan dan kebutuhan yang berbeda[5].

2.2.5. Rapid Application Development

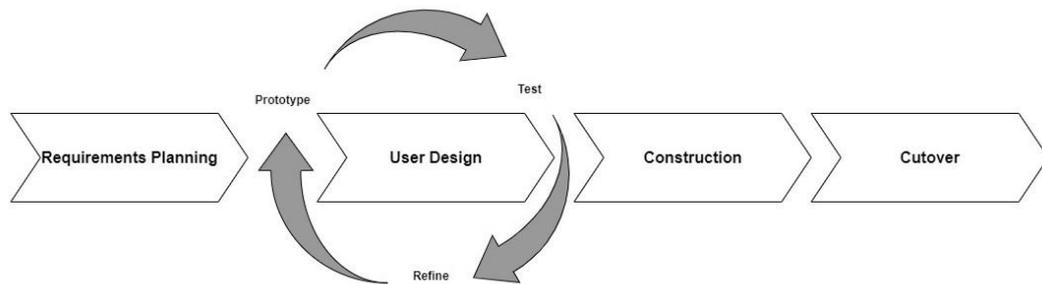
Systems Development Life Cycle (SDLC) mencakup berbagai model yang memfokuskan pada tahapan-tahapan berbeda, seperti perencanaan kebutuhan, desain pengguna, konstruksi, dan implementasi. SDLC adalah serangkaian proses yang terstruktur untuk mengembangkan sistem atau aplikasi perangkat lunak, meliputi tahap-tahap mulai dari pengumpulan kebutuhan pengguna hingga tahap implementasi atau *cutover* ke sistem yang baru. Tiap model SDLC memberikan kerangka kerja yang terorganisir untuk mengelola tahapan-tahapan ini, memungkinkan tim pengembang merencanakan proyek dengan efektif sesuai dengan sifat dan skala proyek yang sedang dikerjakan[13].

Metode *Rapid Application Development* (RAD) adalah suatu kerangka kerja perancangan system yang terdiri dari beberapa tingkat atau tahapan dengan siklus waktu yang singkat, memungkinkan untuk proses yang cepat dan efisien. RAD menekankan pada penggunaan metode yang memungkinkan pengembangan sistem dilakukan dengan iterasi singkat, memungkinkan pengembang untuk merancang, mengembangkan, dan menguji secara berulang dalam periode waktu yang relatif singkat, sehingga mempercepat proses pengembangan aplikasi[18]. Model RAD merupakan pendekatan pengembangan aplikasi yang lebih fleksibel dan adaptif yang dapat menyesuaikan perubahan kebutuhan pengguna serta memastikan peningkatan kualitas sistem dengan cepat dan biaya minimal.

Implementasi dari RAD menitikberatkan pada perencanaan yang cepat serta memusatkan perhatian pada tahapan pengembangan perangkat lunak, meliputi pengembangan, pengujian, dan respons terhadap *feedback*. [19].

Proses pengembangan sistem menggunakan metode RAD terbagi menjadi empat tahapan, yaitu *requirements planning*, *user design*, *construction*, dan *cutover*. Tahapan awal metode RAD yaitu *requirements planning* mengacu pada penemuan masalah pada subjek yang diteliti kemudian didefinisikan cakupan proyek secara jelas untuk menemukan solusi bisnis yang tepat. Tahapan metode RAD selanjutnya yaitu *user design* terdiri dari proses pembuatan *prototype*, kemudian setelah siap maka dilakukan tahap *test* kepada *client* untuk mendapatkan *feedback*, dan terahir *feedback* tersebut menjadi masukan untuk melakukan tahap *refine* yaitu perbakan ulang pada *protoype*. Tahapan *user design* mengacu pada bagaimana hasil dari *requirements planning* sebelumnya digambarkan menjadi design sistem yang nantinya akan diteruskan pada tahap pengembangan selanjutnya. Tahapan berikutnya setelah *user design* adalah *construction* yang mengacu pada beagaimana hasil dari *user design* sebelumnya diimplementasikan untuk dijadikan suatu sistem atau aplikasi yang bisa menjadi solusi dari permasalahan bisnis. Tahapan selanjutnya setelah proses *construction* adalah *cutover* yang mengacu pada pengujian sistem yang telah dirancang pada tahap sebelumnya apakah sudah memenuhi kriteria yang diharapkan[13].

Gambar 2.1 menggambarkan siklus implementasi dan tahapan metode RAD, memperlihatkan pendekatan yang terorganisir dan efisien yang digunakannya.



Gambar 2. 1 *Rapid Application Development*[13]

Gambar 2.1 menggambarkan proses-proses berurutan yang terlibat dalam menerapkan metode RAD, *requirements planning*, *user design*, *construction*, dan *cutover*[13]. Berikut adalah penjelasan dari setiap fase:

1. *Requirements Planning*

Tahap awal dalam metodologi RAD dikenal sebagai *requirements planning*. Tahap ini adalah tahap krusial dari awal proses pengembangan sistem dengan metode RAD. Tahap ini menjadi momen di mana temuan permasalahan, tujuan, dan cakupan proyek didefinisikan secara jelas, sementara analisis mendalam dilakukan untuk memahami kebutuhan yang harus dipenuhi oleh sistem yang sedang dikembangkan. Wawancara menjadi landasan utama untuk mendapatkan informasi dari narasumber terkait detail sistem yang akan dirancang untuk mengidentifikasi tujuan sistem dan menemukan solusi untuk mengatasi tantangan bisnis. Proses ini membentuk landasan yang kuat untuk memastikan bahwa tahap selanjutnya dalam pengembangan aplikasi berjalan sesuai dengan harapan dan kebutuhan yang telah ditetapkan[33][13].

2. *User Design*

Tahap ini menggali lebih dalam terhadap kebutuhan yang diidentifikasi selama tahap analisis. Setiap kebutuhan dijabarkan secara rinci, dan digunakan diagram *Unified Modeling Language* (UML), seperti *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *component diagram* dan *deployment diagram* untuk menggambarkan

interaksi pengguna sistem dan mendefinisikan fungsionalitas utama yang akan dikembangkan[18].

Tahap pengembangan sistem akan dimulai pada fase ini. *Prototype* mewakili semua proses sistem, masukan, dan keluaran. Tahap *protoyping* ini selain menggunakan diagram UML untuk memberikan gambaran alur aplikasi kepada *client* juga dibantu dengan menggunakan *wireframe* yaitu sebuah kerangka awal suatu aplikasi sebelum desain dari antarmuka aplikasi tersebut benar-benar diimplementasikan[34]. Selanjutnya pengujian (*testing*) dilakukan terhadap *prototype* kepada klien untuk memastikan bahwa semua prosesnya benar dan memenuhi harapan klien. Setelah pengujian, *prototype* akan disempurnakan(*refine*) dengan menggabungkan umpan balik dari klien. Tahap penyempurnaan ini penting untuk memastikan bahwa *prototype* menjadi lebih baik sesuai dengan kebutuhan dan harapan yang dinyatakan oleh klien. Setelah melakukan iterasi pengujian dan penyempurnaan, sebuah prototipe final akan disampaikan. Berdasarkan urutan setiap langkah tersebut proses ini menjadi landasan penting dalam menghasilkan solusi akhir yang memenuhi kebutuhan klien dengan tepat untuk menuju proses selanjutnya[35][36].

3. *Construction*

Fokus utama pada fase ini adalah pengembangan sistem menggunakan berbagai alat dan kerangka kerja. Tahap ini melibatkan partisipasi aktif pengguna yang dapat mengusulkan perubahan atau peningkatan seiring dengan perkembangan fase konstruksi[37]. Tahap ini merujuk pada langkah awal dalam mewujudkan sistem yang telah direncanakan sebelumnya. Dimulai dengan proses penyusunan *code* program, yang kerap disebut sebagai *coding* yang tujuannya yaitu mengubah *system design* yang telah dipersiapkan sebelumnya menjadi sebuah aplikasi yang siap diimplementasikan[38].

4. *Cutover*

Fase ini melibatkan pengujian menyeluruh terhadap seluruh sistem yang telah dibangun, di mana setiap komponen sistem harus melalui proses pengujian yang komprehensif menerapkan metode *Black-box Testing*. Hal ini diterapkan untuk mengurangi kemungkinan adanya cacat atau kesalahan dalam sistem yang telah dikembangkan. *Black-box Testing* merupakan suatu metode *testing* perangkat lunak yang memusatkan perhatian pada pemeriksaan terhadap spesifikasi fungsi dari *software* atau perangkat lunak. Fokus utamanya bukanlah pada struktur internal atau kode program, melainkan pada pengujian dari luar untuk memverifikasi bahwa aplikasi beroperasi sesuai dengan yang diinginkan atau diharapkan. Proses *testing* pada tahap ini menjadi penting yang tujuannya memastikan bahwa *system* yang dirancang sudah memenuhi standar kualitas dan bisa beroperasi sesuai apa yang diinginkan[38].

Metode RAD memudahkan dalam perancangan sistem dengan mengikuti tahapan-tahapan yang terdefinisi dengan baik ini melalui pendekatan yang terstruktur dan efisien dalam pengembangan sistem, memungkinkan organisasi untuk menyampaikan solusi perangkat lunak berkualitas tinggi dalam waktu yang lebih singkat[13].

2.2.6. *Black-box Testing*

Black-box testing adalah jenis pengujian yang relatif mudah diimplementasikan. Metode ini melibatkan pengujian input dan output dari setiap fungsi atau fitur yang ada dalam sistem. Proses pengujian ini didasarkan pada skenario pengujian yang terdiri dari serangkaian kasus pengujian, ekspektasi pengujian, hasil pengujian, dan status pengujian. Metode ini memungkinkan penguji untuk tidak perlu memiliki akses atau pengetahuan terhadap *source code* dari sistem yang diuji. Meskipun demikian, pemahaman yang baik terhadap alur kerja sistem yang diharapkan tetap diperlukan. *Functional testing* merupakan sebutan lain yang kerap digunakan untuk menunjukkan jenis pengujian ini karena fokusnya yang lebih kepada pengetesan fungsionalitas sistem secara spesifik[39].

Pengujian *Black-box*, yang juga dikenal sebagai *functional testing* atau pengujian partisi, berfokus terhadap pemeriksaan spesifikasi fungsional suatu *software*. Tujuan utamanya adalah untuk mengevaluasi elemen fungsional pada suatu sistem perangkat lunak. Metode *Black-box* pada pengujian ini menggunakan pendekatan *equivalence partitioning* yaitu sebuah metode yang pengujiannya berdasarkan input pada suatu form sistem aplikasi, dimana setiap menu dikelompokkan berdasarkan fungsinya dengan menghasilkan dua hasil uji yaitu valid dan tidak valid. Tujuan dari pendekatan ini adalah untuk mengidentifikasi fungsi yang salah atau hilang, menemukan kecacatan dalam *interface*, menemukan kesalahan dalam struktur data atau akses database, menemukan permasalahan kinerja aplikasi, dan mengetahui kecacatan dalam inisialisasi dan *output* akhir[40][41].

2.2.7. *System Usability Scale*

System Usability Scale (SUS) merupakan metode pengukuran kepuasan pengguna yang melibatkan pemberian kuesioner formal kepada pengguna setelah aplikasi digunakan yang diciptakan oleh John Brooker pada tahun 1986. SUS digunakan untuk secara cepat mengukur persepsi pengguna terhadap kemudahan penggunaan sistem dengan menggunakan skala *likert*[42]. Instrumen pengujian pada metode SUS menggunakan 10 butir pertanyaan yang terdiri dari masing-masing lima pernyataan negatif dan pernyataan positif. Skala rasio dari setiap pernyataan memiliki nilai satu sampai lima dengan rincian nilai 1 untuk pernyataan sangat tidak setuju, nilai 2 untuk pernyataan tidak setuju, nilai 3 untuk pernyataan netral, nilai 4 untuk pernyataan setuju, dan nilai 5 untuk pernyataan sangat setuju[43]. Berikut ini adalah penjabaran setiap poin instrumen pernyataan pengujian yang akan disajikan pada tabel 2. 2.

Tabel 2. 2 Instrumen Pernyataan SUS Bahasa Inggris[42]

No	Instrumen Pernyataan Asli
1	<i>I think that I would like to use this system frequently</i>
2	<i>I found the system unnecessarily complex</i>

No	Instrumen Pernyataan Asli
3	<i>I thought the system was easy to use</i>
4	<i>I think that I would need the support of a technical person to be able to use this system</i>
5	<i>I found the various functions in this system were well integrated</i>
6	<i>I thought there was too much inconsistency in this system</i>
7	<i>I would imagine that most people would learn to use this system very quickly</i>
8	<i>I found the system very cumbersome to use</i>
9	<i>I felt very confident using the system</i>
10	<i>I needed to learn a lot of things before I could get going with this system</i>

Setelah data kuesioner dari responden berhasil dikumpulkan, maka langkah selanjutnya adalah menghitung nilai dari kuesioner tersebut dengan mengikuti aturan perhitungan dari metode SUS berikut ini.

1. Nilai didapatkan dari skala yang dipilih responden dikurangi satu untuk setiap pernyataan bernomor ganjil.
2. Pernyataan bernomor genap cara menghitungnya adalah dengan menetapkan nilai awal lima kemudian dikurangi dengan nilai skala yang dipilih oleh responden.
3. Nilai SUS didapat dari hasil penjumlahan antara semua nilai yang didapat tadi kemudian dikali 2.5.

Aturan perhitungan nilai SUS ini berlaku untuk masing-masing responden. Setelah semua data dihitung kemudian setiap nilai dari responden dijumlahkan semuanya kemudian dibagi dengan jumlah responden untuk mendapatkan nilai dari rata-ratanya. Berikut adalah rumus untuk menghitung nilai dari hasil pengujian SUS untuk setiap responden akan disajikan pada rumus 2. 1 dan rumus

untuk mendapatkan nilai rata-rata dari perhitungan pengujian SUS disajikan pada rumus 2. 2 [36][38].

$$\begin{aligned} SUS\ Score = & [(Q_1 - 1) + (Q_3 - 1) + (Q_5 - 1) + (Q_7 - 1) + (Q_9 - 1) \\ & + (5 - Q_2) + (5 - Q_4) + (5 - Q_6) + (5 - Q_8) \\ & + (5 - Q_{10})] \times 2.5 \end{aligned} \quad (2.1)$$

Keterangan:

Q = Pertanyaan

$$\bar{x} = \frac{\sum x}{n} \quad (2.2)$$

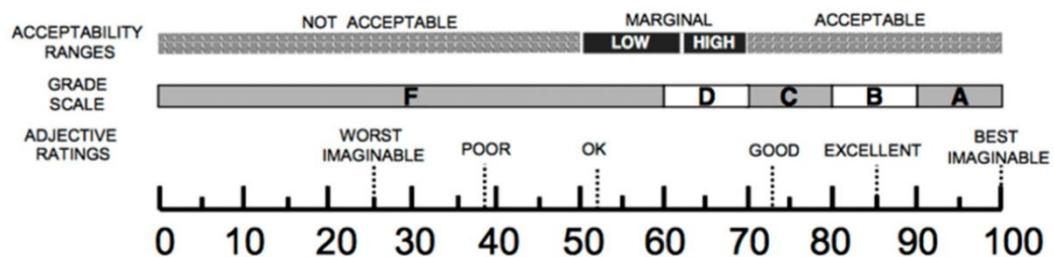
Keterangan:

\bar{x} = Nilai rata-rata

$\sum x$ = Jumlah semua nilai SUS

n = Jumlah seluruh responden

Setelah rata-rata skor dari SUS ditemukan maka langkah selanjutnya adalah menentukan kategori nilai SUS berdasarkan hasil dari rata-rata nilai SUS pada gambar 2. 2 berikut.



Gambar 2. 2 Tingkatan Skor SUS[45]

Gambar 2. 2 menggambarkan bahwa nilai dari SUS dapat disesuaikan dengan kategori yang tertera pada gambar tersebut secara vertikal untuk mengetahui tingkat penerimaan user.

2.2.8. Javascript

JavaScript diperkenalkan pada tahun 1995 oleh Netscape. Pada saat peluncurannya, bahasa ini pertama kali disebut sebagai "LiveScript" dan

difungsikan sebagai bahasa yang cukup sederhana untuk browser Netscape Navigator 2. JavaScript berperan sebagai serangkaian skrip yang bekerja pada dokumen HTML, dan sepanjang sejarah internet, menjadi bahasa *script* pertama yang digunakan untuk web. Tujuan utama dari bahasa ini adalah untuk memberikan tambahan kemampuan pada HTML dengan memungkinkan eksekusi perintah di sisi user, yakni di dalam browser, bukan di server web. Penggunaan JavaScript bergantung pada browser (navigator) yang mengunduh halaman web yang berisi *script* JavaScript, yang ditanam dalam dokumen HTML.[46].

Masa depan JavaScript sangat erat kaitannya dengan perkembangan terus-menerus dari kerangka kerja (*frameworks*) dan perpustakaan (*libraries*). Kerangka kerja populer seperti React, Angular, dan Vue.js terus membentuk cara pengembang membangun antarmuka pengguna (*user interfaces*) dan aplikasi satu halaman (*single-page applications*). Selain itu, perpustakaan dan kerangka kerja baru terus muncul untuk mengatasi kebutuhan khusus, seperti manajemen status, routing, dan rendering di sisi server. Menyimpan informasi terkini tentang tren terbaru dalam kerangka kerja dan perpustakaan JavaScript menjadi hal yang penting bagi pengembang untuk tetap bersaing dalam industri pengembangan web yang terus berubah dengan cepat[47].

2.2.9. Next Js

React adalah proyek JavaScript yang telah mendominasi ekosistem *front-end* belakangan ini. Para ahli dari Facebook merancang dan mengopen-*source*-kan React. Pengembang menulis dan menyusun komponen-komponen antarmuka web dengan React. React memperkenalkan konsep-konsep yang *revolutioner* dan menantang pengembang untuk memikirkan ulang praktik-praktik terbaik. Pengembang web diajarkan selama bertahun-tahun untuk membuat HTML, JavaScript, dan CSS secara terpisah. React merekomendasikan untuk menulis HTML dan CSS dalam JavaScript. Setelah diuji coba, hal ini tidak terlihat begitu aneh. Karena perkembangan pengembangan *front-end* menuju pengembangan berbasis komponen[48].

Next.js merupakan suatu *framework* dari React yang sangat fleksibel yang memungkinkan pembuatan aplikasi web secara cepat. Next.js membutuhkan penggunaan React sebagai perpustakaan (*library*) JavaScript untuk menciptakan antarmuka pengguna (*User Interface*) yang interaktif. Kerangka kerja ini menawarkan kecepatan dan kemudahan dalam mengembangkan aplikasi web, dengan menggunakan React sebagai dasar utama untuk menciptakan pengalaman pengguna yang interaktif dan responsif[49].

2.2.10. Tailwind CSS

Tailwind CSS merupakan sebuah kerangka kerja CSS yang menekankan penggunaan *utility-first*, yang terkenal karena pendekatannya yang revolusioner dalam proses pengembangan tampilan web. Dibandingkan dengan kerangka kerja tradisional, Tailwind memberikan kemudahan dalam menyesuaikan gaya langsung pada elemen HTML, tanpa keharusan untuk melompat-lompat antara dokumen HTML dan file CSS terpisah. Hal ini memungkinkan pengembang untuk langsung merancang dan menyesuaikan tampilan elemen dengan cepat, mengurangi kompleksitas dalam pengelolaan gaya dan mempercepat proses pembuatan tampilan halaman web. Kelebihan lainnya dari Tailwind adalah fleksibilitasnya dalam menyesuaikan *styling*, yang memungkinkan pengembang untuk dengan mudah membuat perubahan dan menyesuaikan tampilan sesuai kebutuhan proyek secara mendalam. Penggunaan Tailwind CSS memungkinkan pengembang dapat lebih fokus pada struktur dan konten dari halaman web yang sedang dibangun, karena Tailwind memberikan alat yang kuat untuk menyesuaikan tampilan tanpa meninggalkan dokumen HTML, sehingga menghasilkan proses pengembangan yang lebih efisien dan responsif[50][51].

2.2.11. Application Programming Interface (API)

API merupakan sebuah antarmuka yang memfasilitasi akses ke aplikasi atau layanan dari suatu program tanpa perlu membangun kembali semua fungsi dari awal. Penggunaan API memungkinkan para pengembang untuk memanfaatkan fitur-fitur yang telah ada pada aplikasi lain dengan lebih efisien. API beroperasi dengan melakukan pemanggilan fungsi melalui protokol HTTP dan menerima

respon dalam bentuk XML atau JSON dalam konteks pengembangan aplikasi. Pengembang dapat mengintegrasikan fungsionalitas dari berbagai aplikasi atau layanan secara lebih fleksibel dan menyederhanakan proses pengembangan aplikasi baru memanfaatkan sumber daya yang telah ada dengan menggunakan API tersebut. Hal ini memungkinkan terciptanya sistem yang lebih terhubung dan terintegrasi di dalam lingkungan teknologi informasi[52].

2.2.12. Midtrans

Midtrans merupakan sebuah *payment gateway* yang menyediakan layanan *online* untuk metode pembayaran dengan keunggulan kecepatan dan kemudahan yang praktis[53]. Midtrans menyediakan berbagai metode pembayaran, termasuk *card payment*, *bank transfer*, *direct debit*, *e-wallet*, dan *over the counter* yang menjadi perantara antara pemilik aplikasi dan institusi keuangan[54]. Keunggulan midtrans yaitu bisa memudahkan integrasi pada suatu aplikasi melalui beberapa *framework* seperti Node Js maupun React Js melalui API yang mudah diintegrasikan[55]. Midtrans juga menawarkan fitur sistem analisis data untuk pengelolaan risiko dan sistem chat kepada pelanggan untuk meningkatkan volume transaksi. Berdasarkan kemudahan dan keunggulannya, Midtrans bisa menjadi solusi yang tepat bagi para pelaku industri *online* dalam meningkatkan operasional dan penjualannya[53].

2.2.13. PostgreSQL

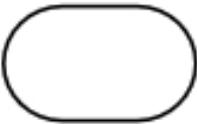
PostgreSQL adalah sebuah sistem informasi manajemen *database* relasional yang dibagikan secara bebas berdasarkan Perjanjian Lisensi BSD. *Database* ini dapat berjalan pada berbagai macam OS besar seperti Linux, Unix, Windows, dan Mac OS X. Kelebihan utama dari PostgreSQL terletak pada sifatnya yang *open-source* dan kemampuannya untuk disesuaikan dengan bahasa pemrograman tanpa memerlukan kompilasi ulang *database*. Salah satu aspek yang menjadikan PostgreSQL unggul adalah manajemen keamanan yang solid terhadap akses pengguna *database*. Selain itu, PostgreSQL dilengkapi dengan beragam fitur penting seperti pemulihan titik waktu, *tablespace*, replikasi asinkron, *savepoints*, dan *backup online*, yang menjadikannya pilihan ideal untuk aplikasi dengan skala

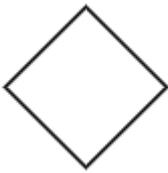
menengah hingga besar dengan kebutuhan penyimpanan data yang beragam[56][57].

2.2.14. Flowchart

Flowchart merupakan gambaran visual mengenai berbagai langkah beserta urutan prosedur dari suatu sistem. Representasi ini membantu analis serta pengembang perangkat lunak dalam menjabarkan struktur aplikasi ke dalam bagian-bagian yang lebih terperinci, memudahkan analisis. Pemecahan masalah dalam aplikasi dapat dilakukan dengan lebih mudah dengan bantuan *flowchart* karena memperjelas logika serta proses penyelesaiannya. Diagram ini umumnya digunakan dalam perancangan aplikasi untuk menggambarkan secara visual bagaimana logika aplikasi tersebut bekerja.[58]. Jadi, tujuan utama dari *flowchart* adalah untuk mengilustrasikan langkah-langkah penyelesaian suatu permasalahan dengan sederhana, terstruktur, dan jelas dengan menerapkan berbagai simbol standar yang terdapat pada *flowchart*[59]. Berikut adalah keterangan atau fungsi setiap simbol dalam *flowchart* yang disajikan pada tabel 2. 3.

Tabel 2. 3 Flowchart [60]

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Mewakili awal atau akhir dari algoritma atau proses.
2		<i>Input/Output</i>	Mewakili input dari pengguna atau output hasil dari algoritma.
3		<i>Process</i>	Mewakili langkah-langkah atau tindakan yang dilakukan dalam algoritma.

No	Simbol	Nama	Keterangan
4		<i>Decision</i>	Mewakili titik di mana suatu keputusan dibuat yang menunjukkan kemungkinan kondisi yang ada.
5		<i>Flow Direction</i>	Mewakili hubungan antara dua simbol yang arahnya menuju simbol tujuan dan biasanya disebut juga <i>connecting line</i> .

2.2.15. UML

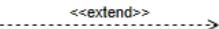
Unified Modelling Language (UML) merupakan suatu bahasa pemodelan yang fungsinya merincikan, menggambarkan secara visual, membangun, dan mendokumentasikan elemen-elemen (*artifacts*) yang terlibat dalam perancangan *software*. *Artifacts* ini bisa berbentuk deskripsi, model, maupun *software* itu sendiri. UML menyediakan berbagai simbol dan notasi yang membantu dalam menggambarkan sistem secara terperinci dari beragam sudut pandang. Penggunaan UML tidak hanya terbatas pada pemodelan perangkat lunak, melainkan juga digunakan secara luas dalam berbagai bidang yang memerlukan pemodelan untuk menjelaskan atau mendokumentasikan suatu sistem. Salah satu komponen kunci dari *Unified Modeling Language* (UML) adalah diagram. Diagram ini merujuk kepada representasi grafis yang menampilkan simbol-simbol dari model elemen yang dirancang sebagai gambaran bagian maupun karakteristik khusus dari suatu sistem[61]. Beberapa diagram yang akan digunakan untuk merancang sistem ini diantaranya:

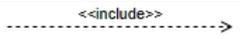
1. *Use Case Diagram*

Use Case Diagram merupakan representasi dari perilaku sistem informasi yang sedang dirancang. Fungsi dari diagram ini adalah untuk mengidentifikasi fungsi-fungsi yang terdapat dalam sistem sekaligus menentukan siapa yang memiliki akses untuk menggunakan fungsi-fungsi

tersebut[62]. *Use case diagram* terdiri dari beberapa simbol yang akan dijelaskan masing-masing fungsinya pada tabel 2. 4 dibawah ini.

Tabel 2. 4 *Use Case Diagram*[63]

No	Simbol	Nama	Keterangan
1	 Actor	<i>Actor</i>	Mewakili entitas eksternal seperti pengguna maupun aplikasi atau sistem lain yang berhubungan dengan sistem yang dibangun.
2	 Use Case	<i>Use Case</i>	Merepresentasikan fungsi atau aktivitas yang dapat dilakukan oleh sistem
3		<i>Association</i>	Simbol ini menunjukkan hubungan antara actor dan fungsi yang diperankan oleh use case tersebut. Garis ini menggambarkan bagaimana actor berinteraksi dengan use case.
4		Ekstensi/ <i>Extend</i>	Simbol ini menunjukkan hubungan antar <i>use case</i> yang menunjukkan bahwa satu use case dapat memperluas atau menambah fungsionalitas dari use case lain pada kondisi tertentu.
5		Generalisasi/ <i>Generalized</i>	Menggambarkan bahwa use case

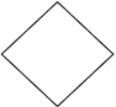
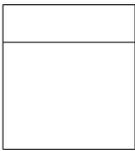
No	Simbol	Nama	Keterangan
			yang lebih spesifik adalah turunan atau merupakan spesialisasi dari use case yang lebih umum
6		<i>Included</i>	Simbol ini menunjukkan hubungan antar <i>use case</i> yang menunjukkan bahwa satu use case membutuhkan fitur dari use case lain untuk menyelesaikan tugasnya.

2. Activity Diagram

Diagram aktivitas (*activity diagram*) adalah representasi visual yang mengilustrasikan alur kerja atau aktivitas dari suatu sistem, proses bisnis, atau menu yang ada dalam perangkat lunak. Diagram ini menampilkan serangkaian aktivitas dalam sistem dan fokus pada aktivitas sistem itu sendiri, bukan pada tindakan yang dilakukan oleh aktor yang terlibat dalam sistem tersebut[62]. *Activity diagram* terdiri dari beberapa simbol yang akan dijelaskan masing-masing fungsinya pada tabel 2. 5 di bawah ini.

Tabel 2. 5 Activity Diagram[63]

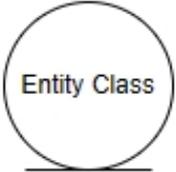
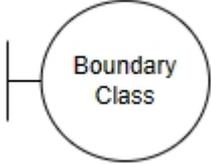
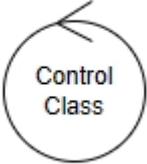
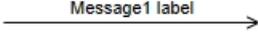
No	Simbol	Nama	Keterangan
1		Status Awal / <i>Initial State</i>	Menunjukkan titik awal dari aktivitas atau proses yang direpresentasikan dalam diagram.
2		Status Akhir / <i>Final State</i>	Representasi dari titik akhir dari aktivitas atau proses dalam diagram

No	Simbol	Nama	Keterangan
3		Aktivitas	Menggambarkan tindakan atau aktivitas yang dilakukan dalam proses. Aktivitas ini bisa berupa operasi, tugas, atau langkah konkret dalam proses yang dimulai dengan kata kerja.
4		Percabangan / <i>Decision</i>	Simbol ini menunjukkan bahwa suatu aktivitas akan mengarahkan pada beberapa aktivitas lainnya.
5		Penggabungan / <i>Join</i>	Simbol ini menunjukkan bahwa aktivitas-aktivitas yang sebelumnya terpisah kemudian bergabung ke satu aktivitas utuh.
6		<i>Swimlane</i>	Simbol ini menjadi pemisah tanggungjawab antar anggota bisnis atas aktivitasnya.

3. *Sequence Diagram*

Diagram urutan (*Sequence Diagram*) merupakan sebuah representasi grafis yang memvisualisasikan serangkaian proses yang dilakukan oleh pengguna dalam suatu sistem informasi. Penggambaran ini didasarkan pada urutan waktu dari setiap tahapan proses yang terjadi. Interaksi antar objek-objek dijelaskan secara terstruktur sesuai dengan urutan waktu di dalam diagram ini. Secara sederhana, *Sequence Diagram* merinci langkah-langkah yang harus diambil untuk mengembangkan suatu sistem yang sesuai dengan gambaran *use case diagram*[64]. *Sequence diagram* terdiri dari berbagai macam simbol yang akan dijelaskan pada tabel 2. 6.

Tabel 2. 6 *Sequence Diagram*[63]

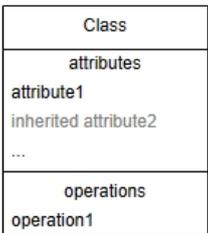
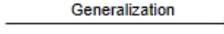
No	Simbol	Nama	Keterangan
1	 Actor	<i>Actor</i>	Mewakili entitas eksternal seperti pengguna maupun aplikasi atau sistem lain yang berhubungan dengan sistem yang dibangun.
2	 Entity Class	<i>Entity Class</i>	Mewakili entitas data atau bisnis dalam sistem. Ini bisa berupa tabel database, objek, atau struktur lain yang menyimpan data.
3	 Boundary Class	<i>Boundary Class</i>	Mewakili batas atau antarmuka antara sistem dan entitas eksternalnya. Biasanya mencakup komponen antarmuka pengguna.
4	 Control Class	<i>Control Class</i>	Mewakili kelas atau komponen yang bertanggung jawab atas pengendalian dan koordinasi alur sistem. Umumnya berisi logika bisnis.
5		<i>A focus of control & a life line</i>	Mendesripsikan lokasi diawali dan selesainya suatu <i>message</i> .
6	 Message1 label	<i>Message</i>	Mewakili komunikasi atau interaksi antara elemen yang berbeda dalam sistem. Ini dapat berupa pemanggilan metode, pertukaran data, atau bentuk komunikasi lainnya.

4. Class Diagram

Diagram kelas (*Class Diagram*) merupakan suatu representasi visual untuk mengilustrasikan struktur basis data dan kelas objek. Pemodelan ini

bertujuan untuk memberikan kelas pada setiap basis data, sehingga memberikan atribut sebagai proses alur data. Diagram kelas mengilustrasikan arsitektur sistem berdasarkan perspektif definisi antar kelas yang dirancang atas sistem yang dibangun[64]. *Class diagram* memiliki beberapa simbol yang akan didefinisikan fungsinya pada tabel 2.7.

Tabel 2.7 *Class Diagram*[63]

No	Simbol	Nama	Keterangan
1		Kelas / <i>Class</i>	Representasi dari suatu entitas atau objek dalam sistem yang sedang dimodelkan.
2		Antarmuka / <i>Interface</i>	Merupakan kumpulan dari metode-metode yang dideklarasikan tanpa implementasi.
3		Asosiasi / <i>Association</i>	Asosiasi menggambarkan bahwa kelas satu dapat memiliki akses ke kelas lainnya.
4		Asosiasi Berarah / <i>Directed Association</i>	Bentuk khusus dari asosiasi yang memiliki arah atau orientasi tertentu. Ini menunjukkan arah spesifik dari keterkaitan antara kelas-kelas tersebut.
5		Generalisasi	Konsep pewarisan di mana suatu kelas (<i>subclass</i>) dapat mewarisi atribut, perilaku, dan hubungan dari kelas lain yang lebih umum (<i>superclass</i>).
6		Kebergantungan / <i>Dependency</i>	Hubungan antara kelas yang menunjukkan bahwa satu kelas menggunakan layanan atau fitur dari kelas lain.

No	Simbol	Nama	Keterangan
7		Agregasi / <i>Aggregation</i>	Menggambarkan keterkaitan antara dua kelas, di mana sebuah kelas (" <i>whole</i> " atau keseluruhan) memiliki bagian-bagian atau komponen-komponen (" <i>parts</i> " atau bagian-bagian tersebut).